Sentence-based Summarization of Scientific Documents

The design and implementation of an online available automatic summarizer

W.T. Visser
Department of Computing Science
University of Groningen
w.t.visser@student.rug.nl

M.B. Wieling
Department of Computing Science
University of Groningen
m.b.wieling@student.rug.nl

Abstract* In Edmundson (1969) four features are used: the title feature, the cue word feature, the location feature and the word frequency feature. The frequency method Edmundson applied used the frequency of relevant words (frequency larger than a certain threshold and not being a common word) and assigned a score to each sentence based on the frequency of the relevant words in the sentence. Because of these reasons the method of Luhn (1958) is chosen as the preferred method for calculating the frequency score of a sentence. A sentence-based automatic summarization system has been developed, which benefits from a mixture of ideas founded in the early days of automatic summarization.

Keywords: summarization, summary, scientific documents, sentence, cue phrase, key, title, location, query, tagging

1 Introduction

Since the introduction of personal computers digital information has become of ever increasing importance. The availability of digital information has exploded even further with the introduction of the Internet. To quickly and easily find relevant documents corresponding to users' needs, reading complete, possibly large texts is inadequate. It is the automatic summarization system filling this niche.

Summaries are to reduce the complexity and length of texts, while retaining the most important information. Summaries can be represented by a hierarchy of headings with corresponding most important sentences, or by the most important sentences only. As the quality of a summary can be measured by the amount of relevant information present (and omitted), the length of the summary presents an issue. The higher the compression ratio, the more probable important information was omitted. A journal paper of 20 pages which is summarized in only half a page, has a compression ratio of 2.5%, while a newswire article compressed to the same size probably has a compression ratio between 20% and 30% and therefore will contain more relevant information.

In the early days of automatic text summarization (hereinafter denoted ATS) researchers referred to abstracts and extracts, i.e. selection of text portions, as similar notions. To date there is a common agreement on the difference between these two concepts. Scientific papers usually already consist of a summary, written by the author, known as an abstract. This special form of summary, which represents the contents of a text not by extracted sentences but by sentences generated from the main concepts of the paper, introduces a major problem to be solved in all ATS systems: how to catch the main concepts in just a few sentences, while attaining a semantically and syntactically correct text? While many researchers have offered techniques to tackle this problem, it has still to be solved. Most problematic being the resolvement of dangling anaphores and the assessment of related clauses. See also Paice (1990) for more information.

Although it are these abstracts that trigger the essential problems in ATS, our main goal is to capture the aspects that form the basis of a summary. We are interested in the - individually and relatively - qualitative performance of early summarization systems. As a result, our focus will be on the method proposed by Edmundson (1969), combined with some ideas of Luhn (1958) and the extension from single word to cue phrase significance as used in Teufel & Moens (1997). We also experimented with generating summaries based on simple queries. A qualitative evaluation of these items will be discussed later.

An overview of previous work, dating back from the early 1950's up to modern approaches covering sophisticated methods as marginal relevance, multi-document summarization and rhetorical structures, will be offered in the next section. We will then introduce our approach of summarization by text selection followed by an elaboration, addressing the architecture and capabilities of the automatic summarization system we developed. Experiments and results are evaluated in the fifth section, followed by a discussion. The last section offers the conclusions.

2 Previous work

Since the introduction of the field of ATS by Luhn (1958), many approaches to the resolvement of the problems in this

*This abstract has been generated by the automatic summarizer developed during this project, which is available at *http://home.hccnet.nl/m.b.wieling/autosummarizer.html*

field have been proposed. As to its ill-definement it still is an open field of science in which progress is spurious. In this section we will offer a brief overview of approaches adopted by the field.

2.1 Traditional summarization systems

Luhn (1958), upon introduction of ATS to the science community, proposed a single statistical measurement still in use to date: the term frequency (TF). Picked up by Edmundson (1964), addressing the problems in this new field, a first significant contribution to the scientifical solution of ATS was proposed by Edmundson (1969). In his article he introduced the notion of sentence position, cue words and title words. Although his work is regarded as an important contribution differences with the work of Luhn (1958) can be found, which we will discuss later.

Kupiec *et al.* (1995), in agreement with Edmundson, elaborated on these features by extending it with a naive Bayes classifier. Further use of statistics has been proposed by Conroy & O'Leary (2001), who introduced the HMM for text selection. Another Machine Learning approach, K-means clustering, was introduced to the field of ATS by Nomoto & Matsumoto (2001) using it as a tool for retrieving the text diversity.

While the previously discussed approaches are summary extraction methods, Hovy & Lin (1997) aimed at constructing an abstract. They described the task of ATS by a {*identification,interpretation,generation*}-triple using sentence position (Lin & Hovy, 1997), cue phrases and topic identification. As a preprocessing step part-of-speech tagging, an approach originating from the field of natural language processing, has been adopted.

2.2 Cohesion and coherence

In order to locate the salient elements of a text, various methods have been proposed. Among them, cohesion focusses on some of the intrinsic properties of texts. As stated by Mani *et al.* (1998),

> cohesion involves the relations between words or referring expressions, which determine how tightly connected the text is

and hence functions as a way to 'hang together' a text (Halliday & Hasan, 1976). Although cohesion is a volatile concept, not easily captured by linguistic means, approximations have been the focus of research; the most frequently discussed method being the lexical cohesion. Morris & Hirst (1991) introduced a method computing lexical chains (lists of connected phrases describing a distinct subject). This method lacked an implementation due to absence of the required digital thesaurus at that time. An improved version has been proposed by Barzilay & Elhadad (1997), which was even further elaborated upon by Silber & McKoy (2002).

Somewhat independent of cohesion, following the definition by Mani *et al.* (1998),

> coherence reflects the deliberate organization of the text by the author in terms of a hierarchical structure to achieve particular argumentative goals.

It has been suggested by Marcu (1997b) that cohesion can be used for coarse-grained segmentation (selecting significant, paragraph-size, portions of text) delivering a reliable sub-text for coherence to be applied as a more fine-grained segmentation (retrieving the salient clauses). This correspondence has been further investigated in a contrasting research by Mani *et al.* (1998).

The theory of coherence can be used as an analyzing tool of text structure, one of the most widely used theories being the rhetorical structure theory (RST) (Mann & Thompson, 1988). Recent interest has been shown by Marcu (1996), formalizing the ambiguous theory. Continuing his preliminary research, Marcu (1997a) proposed a method composing these discourse structures to text summaries. His complete work can be found in (Marcu, 1997b). A combination of RST with relevance in summarizing scientific articles has been the focus of Teufel & Moens (2002).

2.3 Multi-document summaries

The growth of on-line information increasingly requires reliable methods for fast information retrieval. It is this need that merge together the fields of ATS and conventional IR. Introducing summarization mechanisms to offer users quick means of assessing relevant documents has been a major improvement on standard search techniques. As multi-document ATS systems operate on large amounts of texts, the condensation of information is larger than ever. Hence, determining phrase relevance has become an even more challenging task. Maximal marginal relevance (MMR), a method introduced by Carbonell & Goldstein (1998), selects phrases relevant to a user's query while being minimally relevant to phrases already present in the constructing summary. Extending MMR to the area of multi-document summarization Goldstein *et al.* (2000) proposed the query-dependent multi-document MMR (MMR-MD), incorporating time sequence and document clusters among other features. Ganapathiraju (2002) extended this work with improved passage clustering based on cluster granularity. Focussing more on document and sentence clustering, Radev *et al.* (2004) have proposed an alternative to MMR(-MD), called CSIS, designed for query independent summarization. Although following a different approach, they both are strongly related to the TF(-IDF) heuristic, alike most ATS systems.

3 Summarization model

As already mentioned in Luhn (1958) and Edmundson (1969), a summary of a document can be automatically generated by selecting the most relevant sentences. The relevance of a sentence is based on specific (statistical) features. In Luhn (1958) the word frequency is used as the only feature. In Edmundson (1969) four features are used: the title feature, the cue word feature, the location feature and the word frequency feature. Our summarization model is based highly on Edmundson (1969) with some modifications.

3.1 Significant text characteristics

The following significant text characteristics are used to generate the summary:

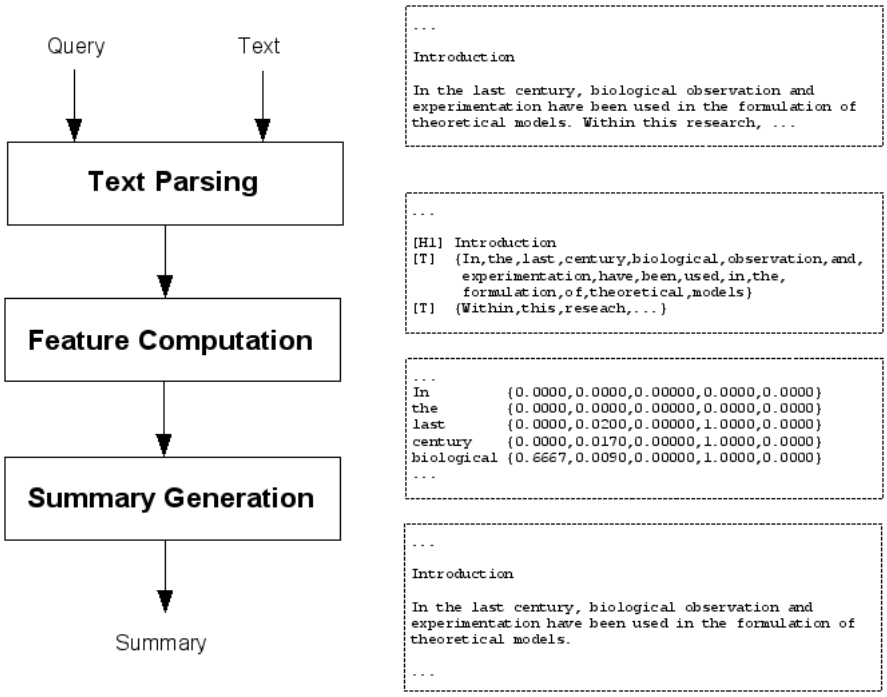

Figure 1: *Architecture of the summarization system.*

- Word Frequency method
- Cue Phrase method
- Location method
- Title method
- Query method

In the following sections an elaboration is given on these statistical features.

3.2 Word Frequency method

The first feature discussed here is word frequency. As mentioned by Luhn (1958) word frequency is a useful measure for determining word significance.

The frequency method Edmundson applied used the frequency of relevant words (frequency larger than a certain threshold and not being a common word) and assigned a score to each sentence based on the frequency of the relevant words in the sentence. This method has one major drawback, which can best be illustrated by an example. Suppose a text about Biology is summarized, which contains the relevant word 'cell' five times as often as any other relevant word. Assume one sentence contains only the relevant word 'cell', while the other (more significant) sentence contains four other relevant words. In this case the first sentence will be selected in favor of the more significant second sentence.

The method Luhn (1958) used did not take into account the exact frequency of the separate words, it only distinguished significant words (frequency larger than a certain threshold and not being a common word) from non-significant words. Furthermore it used the relative position of significant words, which also was not used in Edmundson (1969). We support the idea that a sentence is more relevant when significant words appear in close vicinity of each other.

Because of these reasons the method of Luhn (1958) is chosen as the preferred method for calculating the frequency score of a sentence.

3.3 Cue Phrase method

The Cue Phrase method is based on the assumption that the relevance of a sentence is based on the presence of certain pragmatic phrases. For example, a phrase which indicates positive relevance is 'the conclusion of this paper', while a phrase which indicates negative relevance of a sentence is 'for example'.

Edmundson introduced the Cue method in 1969. In his Cue Word method, he used single words and splitted these words into three classes (bonus words, null words and stigma words). Whenever a sentence contained a stigma word, the sentence was given a negative cue score. A positive cue score was given when no stigma words and one or more bonus words were present in the sentence. The null words did not influence the cue score of a sentence.

The Cue Phrase method of Teufel & Moens (1997) extended Edmundson's approach by using phrases instead of words, making the method more flexible, and defining 5 classes instead of three (2 extra bonus phrase classes). Three bonus lists facilitated the idea that a large group of phrases was important to the relevance of a sentence, but one phrase was more important than the other. In this way the cue score of a sentence was higher if the phrase occurred in a more important bonus list.

3.4 Location method

The Location method is based on the assumption that topic sentences have the tendency to occur very early or very late in the document and its paragraphs. The Location method of Edmundson (1969) works in the following way. A positive score is given to sentences occurring in the first and last paragraph of the document, as well as sentences occurring first and last in each paragraph. For each location a different score can be assigned. Finally, the score of a sentence is increased if it appears below a certain heading (e.g. like 'conclusion').

3.5 Title method

The central idea behind the title method is that the author conceives the title as circumscribing the subject matter of the document (similarly for headings versus paragraphs). The Title method of Edmundson (1969) assigns a positive score to a sentence based on the occurrence of its words in the document title, or in one of its headings.

3.6 Query method

The Query method is a simple method to tailor the summary to a user specified subject. Sentences which match the specified query will get a higher score than sentences which do not match the specified query.

3.7 Feature score combination

To obtain the final score of a sentence, the five features are each given a certain weight. In this way a greater significance can be given to one feature over another. For example, to simulate Luhn's text extraction method the weights of all features, except the Key feature, is set to 0.

To display the summary, the selected number of highest scoring sentences are displayed in the order in which they appear in the source document (with or without their headings).

4 Design of the summarizer

Based on the model discussed in the previous section we will now focus towards the architecture and capabilities of the automatic summarization system we developed.

Input text, possibly containing multiple (levels of) headings, is initially parsed by the preprocessing unit splitting it into words while maintaining the structure of the input. The statistical features, as described in the previous section, on which the summary will be constructed are computed next. Finally, the summary is generated based on the computed feature values of the individual words (and, consequently, the sentences). An overview of this sequential process is illustrated in figure 1.

4.1 Preprocessing

As the generation of a summary will depend on the computed feature values of each sentence, and these sentence values depend on the feature values of the individual words in these sentences, the preprocessing unit should split the text into word groups while attaining the text sentence structure. However, as the Location method requires knowledge about the position of each sentence in a paragraph and the position of each paragraph in the complete text, a simple sentence and word split will not suffice. Furthermore, the Title method requires the presence of a structure containing heading-textblock pairs. Hence, a proper parsing strategy obeying these requirements is needed which will be described in the following sections.

Table 1: *Demonstration of the self-heading relation. Headings are annotated by [Hn]; [Tn] denotes a textblock. Corresponding values of n denote a self-heading relation.*

[H0] **MULTIPLE LEVELS OF HEADINGS**

[H1] **1. Motivation**
[T1] The following section discuss the advantages of the use of multiple headings.
[H2] **1.1 Text structure**
[T2] Multiple levels of headings yield a better understanding of the text's contents.
[H3] **1.2 Search strategy**
[T3] Multiple levels of headings facilitate a better manual search strategy.
...
[H4] **2. Conclusion**
[T4] The use of multiple levels of headings is advisable.

Text structure

Scientific texts generally consist of multiple levels of headings. With each heading, except for the title, a block of text can be associated which is located directly below this heading. In such a relation the heading will be denoted the *self-heading* of the corresponding block. Consider for example the artificial text shown in table 1. While heading [H2] is the self-heading of text [T2], heading [H1] is not as it is not located at the same level as [T2] in the structure of the text.

Heading identification is two-fold: 1) either a heading is a single sentence textblock separated from other textblocks by empty lines, or 2) a heading is tagged as such by an enclosing <Hn>-</Hn> pair, where n denotes the level of the heading in the text structure; a title has level 0 and should be located at the start of the text. All non-tagged single-sentence textblocks are processed as level one headings.

To construct the text structure (more precisely, the text structure tree) a top-down recursive scan on heading tags is performed. Each time a heading is detected its corresponding text (optionally spanning multiple textblocks and lower level headings) is processed. For a leaf heading its level and heading parent (the heading one level up in the tree) is stored. The textblock is annotated with its self-heading. The result of this parsing is illustrated in figure 2.

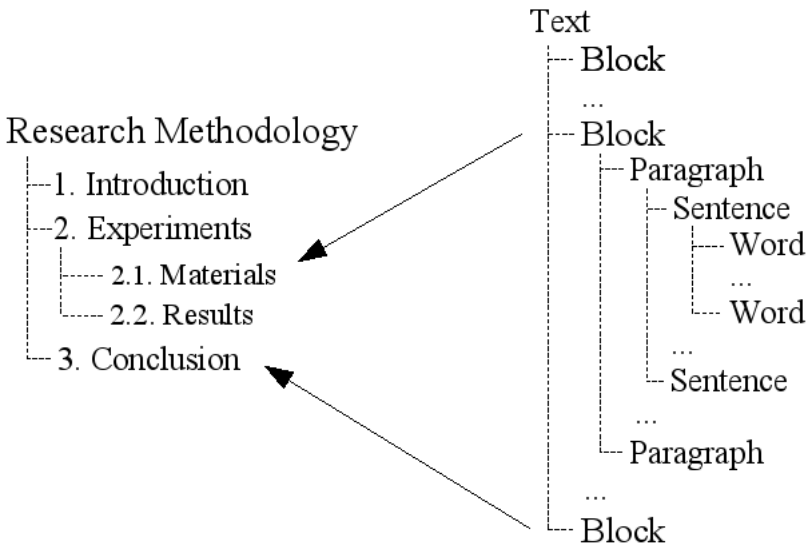

Figure 2: *Diagram of the heading tree of a typical text (left) and text structure tree (right) after parsing. In the text structure tree a single block is fully expanded, showing elements up to word level. A textblock points towards its self-heading. Each heading contains a pointer towards its parent heading.*

Paragraph retrieval

Within a textblock multiple paragraphs can be present, of which the location relative to the offset of the textblock is required for the Location method. Retrieving these paragraphs is done by parsing the textblocks, requiring subsequent paragraphs to be separated by an empty line.

Sentence splitting

The summarization system operates on word level, computing feature values for each of them, and generates a summary by selecting sentences based on the cumulative weighted feature values of its words. First, to obtain the sentences, paragraphs are scanned for sentence delimiters. The following heuristics are used to determine a sentence end:

- A sentence ends with one or more points, exclamation marks and/or question marks.
- Sentence delimiters are optionally followed by an ending quotation mark, e.g. in the following text: *Mandela once said: "I dream of an Africa which is in peace with itself."*
- The final delimiter should be followed by one or multiple white spaces (space, tab, newline, etc.).
- The first word of the following sentence should start with a capital. For Dutch texts the capital could be preceded by an abbreviated genitive word, constituting of an omission mark and a lower-case letter (e.g. *'s* in *'s Avonds*). The first word of a sentence can also be a value.
- The sentence delimiter should not be part of an abbreviation (such as *i.e.*). In order to detect abbreviations a simple abbreviation dictionary is consulted.

Following the sentence split, the individual words are extracted from the sentences by scanning the sentence text on white spaces. A word is defined as a sequence of characters ([A-Z], [a-z] and/or [0-9]). Hence, white spaces delimit a word. Omission marks are not considered words.

Tagging

As the text parser is far from sophisticated, lacking statistical and grammatical methods, and no dictionary can ever contain the complete lexicon, meta-tags are introduced to influence the result of the system. The tags available in the system are:

- <**H***n*>**...**</**H***n*> Forces a text enclosed by these tags to be a heading. The value *n* denotes the level of the heading and should not be more than one level lower than any other heading level. A level 0 tag denotes a title, level 1 tag the main section headings and lower level headings tags corresponding subsections. Note that level 0, and possibly level 1, is implicitly present in the structure. The main advantage of the heading tag is to set a multi-sentence text as heading, which is otherwise parsed as normal text.
- <**T**>**...**</**T**> Forces a text enclosed by these tags to be non-heading text. Its advantage is to parse a single-sentence paragraph as text rather than as heading.
- <**NS**> Indicates that no sentence split should be performed. This tag should be placed directly after (a sequence of) sentence delimiters. It offers the user the capability to ignore a sentence split due to an unrecognized abbreviation.
- <**S**> Forces a sentence split. This is useful when a sentence ends with an abbreviation or a value.
- <**F**> Forwards a sentence directly to the summary generation method, ignoring feature computation and gaining maximum weight. This proves useful when one prefers to include author information, which generally

consists of only a few significant features and hence is unlikely to be included in the summary automatically.

To make tagging more resilient to human errors dangling heading tags are resolved by scanning a paragraph for opening and closing heading tags. If only a opening heading tag is present, and hence would result in a parsing failure, a closing tag is introduced at the end of the paragraph. Similarly, a opening tag is inserted at the beginning of the paragraph when only a closing heading tag has been found.

4.2 Feature computation

The automatic summarizer uses the five features mentioned earlier to generate the summary. It is possible to set the relative weight of each feature. These weights are normally positive, but they can also be negative or 0 (which will result in not using the feature).

- **Cue feature:** The occurrence of certain phrases in the document influences the cue value of the sentence. The phrases are **specifically** tailored for scientific papers. They contain sentences like 'our studies have indicated', which are specific to scientific papers. Since the Cue lists are mere text-files, it is possible to tailor them for different languages (now Dutch and English are supported) or specific fields.
- **Key feature:** The frequency of each uncommon word is counted and the sentence gets a score based on the number of high-frequent words. Note that common words like 'that' or 'and' are filtered out by means of a stop-list. The minimal frequency of a word to be marked as a relevant word can be set by changing the threshold (as a real-valued percentage of the total number of non-heading words). The absolute threshold value (number of words) can also be obtained, as well as the total number of words of the source text. Two methods are available for calculating the key value. The first is Edmundson's key method, which assigns a score to each word equal to its frequency. The key score of the sentence is calculated by summing the scores of its words. The second is Luhn's key method, which does not use the exact frequency of each word, but treats each significant word in the same way. In contrast to Edmundson's method, it also takes the relative position of significant words into account. In Luhn's method a maximum number of non significant words (n.s. words) is specified. Within a sentence a range of words is selected which has a significant word at the beginning and end and doesn't have more non-significant words in this range than is specified. The value of a sentence is then calculated by taking the square of the number of significant words and dividing them by the total number of words in the range, e.g. for the sentence:

  ```
  * - - - [* - * * - - * - - *] - - *
  ```

  With `*`: a significant word, `-`: a non-significant word, and `[...]`: the range with (in this case) max 5 n.s. words. Here the frequency score would be $5 * 5/10 = 2.5$.
- **Title feature:** Each word in the sentence gets a score based on its occurrence in the title or in one of its headings. The title value of the sentence is calculated by summing the scores of its words. The relative weight of a word occurring in a heading or title can be specified.
- **Location feature:** Sentences get a score based on their location (paragraph initial, paragraph final, in the first paragraph or in the last paragraph) and their occurrence beneath certain headings (like 'conclusions' or 'introduction'). The relative weights of each of the four locations can be specified.
- **Query feature:** The query score of a sentence is calculated by matching the sentence to the user specified query (which may be empty). The query is not case sensitive. Sentences which match the query are more likely to appear in the summary. The syntax of the query is as follows: each word which is entered must appear in the sentence, unless the OR keyword is placed between them. Note that nesting is not possible, so (keyword1) OR (keyword2) means: "(keyword1)" OR "(keyword2)". Words in the stop-list, like 'and' are normally ignored, unless the '+' is typed before them. To specify a word which must not appear in the sentence, the sign '-' should precede the word. Exact phrase matches can be specified by quoting the phrase.

Besides the feature specific settings, there are several other options. The most important option is to set the source language of the document (Dutch or English). Also the size (as a real-valued percentage of the size of the source document) of the summary should be entered. Furthermore it is possible to specify if the resulting summary should include the headings, or only consists of the selected summary sentences.

4.3 Summary generation

Based on the specified parameters and feature weights, the summary will be generated and can be copied to the system. The $n\%$ most relevant sentences (i.e. the sentences with highest overall feature weight) are selected as summary sentences, where n is the size of the summary relative to the number of sentences in the input text.

5 Evaluation

Many summarization evaluation methods have been proposed since the uprise of automatic summarization techniques, a recent overview of which is given by Mani (2001). These methods can be roughly classified into two categories: *intrinsic* and *extrinsic* evaluation. Intrinsic evaluation tests the summarization system on itself, whereas extrinsic evaluation is concerned with testing how the summarization effects the completion of some other task (e.g. a human assessing a document's relevance). In this paper the qualification of the performance of the summarization system is measured by an intrinsic evaluation method.

5.1 Corpus and methodology

The evaluation corpus is a collection of 9 Dutch papers and reports from the field of robotics. All documents were

Inleiding

Binnen de robotica bestaat het probleem dat je tijdens het navigeren wilt weten hoe groot de afstand is tussen de robot en het object waar je langs rijdt. Tijdens ons experiment gaan we ervan uit dat de robot degene is die beweegt en dat de omgeving stil staat. Binnen de computer vision kan een optical flow field berekend worden van een camerabeeld, waar vectoren in staan die aangeven of de bijbehorende pixel(groep) beweging vertoont ten opzichte van het vorige camerabeeld.

Figure 3: *Introductory part (corresponds to heading 'Inleiding') of the summary generated by Luhn's Key method (n.s = 4); sentences overlapping the corresponding summary part generated by Edmundson's Key method are omitted.*

Inleiding

Het systeem stelt de robot in staat de afstand cq. snelheid van het object te bepalen waar de robot naar kijkt.

Figure 4: *Introductory part (corresponds to heading 'Inleiding') of the summary generated by Edmundson's Key method; sentences overlapping the corresponding summary part generated by Luhn's Key method are omitted.*

fully tagged to address possible parsing problems. Tables, figures, equations, captions and non-significant appendices were omitted. References to these elements have not been altered.

A full intrinsic and extrinsic evaluation is beyond the scope of this project. Although we acknowledge that, for summary qualification, independent judges (not having any knowledge about the background of the summaries) are required the summary comparison is based on sentence relevance assessment by the authors. Hence, this evaluation should not be considered statistically meaningful; it serves merely as an initial comparison.

In the following sections, the length of each summary have been set to 10% of its corresponding text. Headings are included in the summary.

5.2 Luhn versus Edmundson

In order to carry out a comparative study on the results of Luhn's Key method (with a maximum of 4 non-significant words) and Edmundson's Key method we have generated summaries on these methods by setting all feature weights, except the key weight, to zero. A single, representative document (see appendix A) forms the basis of this comparison. The summaries by Luhn's method and Edmundson's method can be found in appendix C and B, respectively.

It is the authors' opinion that, in agreement with the expectations, Luhn's Key method yields the best result. As an example, consider the part of the summary discussing the theory (i.e. corresponding to the heading 'Inleiding'). For ease, the non-overlapping sentences in both summaries have been displayed in figure 3 and 4. Independent of the full contents of this document the displayed part of the summary generated by Luhn's Key method is preferred, even when considering fact that the number of sentences in this summary part is three times as large as in the result of Edmundson's Key method. It, however, causes other parts of the result of Luhn's Key method to have worse results (cf. the text of the heading *Theorie*). Nevertheless, normalized over multiple documents Luhn's Key method yields a more profound result.

We have experimented with a gold standard summary (see appendix D), to which the summaries generated using the two Key methods have been compared. The gold standard is a summary based on all features except the Key method and hence should not be biased on Key words or phrases. Unfortunately, comparing the summaries generated using the two Key methods with the gold standard, hardly any sentences (10% on average) coincide. The summaries are regarded too distinct to perform a meaningful comparison.

5.3 Query effects

The effect by a query can best be demonstrated on the gold standard summary in appendix D. Using a query weight of 2 (other weights are unaltered), the aim is to introduce the very first sentence from the source text into the summary. The query word 'navigeren' is used to obtain the desired result. The resulting summary can be found in appendix E, which shows two sentences to be introduced relative to the gold standard, namely:

Binnen de robotica bestaat het probleem dat je tijdens het navigeren wilt weten hoe groot de afstand is tussen de robot en het object waar je langs rijdt.

and

Bijen maken zowel gebruik van landmarks als van dead reckoning tijdens het navigeren (Iida en Lambrinos).

6 Discussion

No solid evaluation on the relative performance of Luhn's and Edmundson's Key method has been carried out. But based on results of our experiments, the idea of Luhn's Key method outperforming Edmundson's Key method has been strengthened. We are confident in finding consolidating proof if further research is to be conducted.

The effect of the use of queries to influence the result of the summarization is, as expected, the inclusion of the aimed sentence into the summary. Side effects, however, are other (possibly non-relevant) sentences to be included as well. Furthermore, due to addition of a new sentence to the summary, another sentence has to be omitted. As this

will be the sentence with the lowest feature value, it can be expected to have a minor effect on the quality of the summary.

No experiments on the general quality of the summary have been conducted. Hence, no comparison with other automatic summarization systems can be made. We acknowledge that further evaluation is required in order to converse about this topic.

7 Conclusions

A sentence-based automatic summarization system has been developed, which benefits from a mixture of ideas founded in the early days of automatic summarization. More specifically, it is possible to generate summaries using features described by Edmundson (1969). An extension towards the Key feature of Luhn (1958) has been made, as well as towards Cue phrases introduced by Teufel & Moens (1997) instead of the original Cue words. Finally, the option of using queries to generate the summary has been added.

The automatic summarizer has been developed as a final assignment in the course *Natural Language Processing* lectured by the Computational Linguistics group at the University of Groningen. The summarizer is online available at *http://home.hccnet.nl/m.b.wieling/autosummarizer.html*.

Acknowledgements

We would like to thank Simone Teufel and Marc Moens for making available their English cue-phrases list and stoplist. We would also like to thank Roeland Ordelman for supplying his Dutch abbreviation list. We thank Gertjan van Noord for his support during this project.

References

Barzilay, R., & Elhadad, M. 1997. Using lexixal chains for text summarization. *Pages 10–17 of: Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97)*.

Carbonell, J., & Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Pages 335–336 of: Proceedings of 21st annual international ACM SIGIR conference on Research and development in information retrieval*.

Conroy, J.M., & O'Leary, D.P. 2001. Text summarization via hidden markov models. *Pages 406–407 of: Proceedings of the 24th annual ACM SIGIR conference on Research and development in information retrieval*.

Edmundson, H.P. 1964. Problems in automatic abstracting. *Communications of the ACM*, **7**(4), 259–263.

Edmundson, H.P. 1969. New methods in automatic extracting. *Journal of the ACM*, **16**(2), 264–285.

Ganapathiraju, M.K. 2002. *Relevance of cluster size in MMR based summarizer: a report*. Advisors: Carbonell, J. and Yang, Y.

Goldstein, J., Mittal, V., Carbonell, J., & Callan, J. 2000. Creating and evaluating multi-document sentence extract summaries. *Pages 165–172 of: Proceedings of the 9th international conference on Information and knowledge management*.

Halliday, M., & Hasan, R. 1976. *Cohesion in English*.

Hovy, E., & Lin, C. 1997. *Automatic text summarization in SUMMARIST*. In Mani and Maybury (1997).

Kupiec, J., Pedersen, J., & Chen, F. 1995. A trainable document summarizer. *Pages 68–73 of: Proceedings of the 18th annual ACM SIGIR conference on Research and development in information retrieval*.

Lin, C., & Hovy, E.H. 1997. Identifying topics by position. *Pages 183–290 of: Proceedings of the Applied Natural Language Processing Conference (ANLP-97)*.

Luhn, H.P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, **2**, 159–165.

Mani, I. 2001. Summarization evaluation: an overview. *In: Proceedings of the NTCIR Workshop 2 Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization*.

Mani, I., Bloedorn, E., & Gates, B. 1998. Using cohesion and coherence models for text summarization. *Pages 69–76 of: Proceedings of AAAI Spring Symposium on Intelligent Text Summarization*.

Mann, W.C., & Thompson, S.A. 1988. Rhetorical structure theory: towards a functional theory of text organization. *Text*, **8**(3), 243–281.

Marcu, D. 1996. Building up rhetorical structure trees. *Pages 1096–1074 of: American Association for Artificial Intelligence, Vol 2*.

Marcu, D. 1997a. From discourse structures to text summaries. *Pages 82–88 of: Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*.

Marcu, D. 1997b (December). *The rhetorical parsing, summarization, and generation of natural language texts*. Ph.D. thesis, University of Toronto, Toronto, Canada.

Morris, J., & Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, **17**(1), 21–48.

Nomoto, T., & Matsumoto, Y. 2001. A new approach to unsupervised text summarization. *Pages 24–36 of: Proceedings of the 24th annual ACM SIGIR conference on Research and development in information retrieval*.

Paice, C.D. 1990. Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing & Management*, **26**(1), 171–186.

Radev, D.R., Jing, H., Styś, M., & Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, **40**, 919–938.

Silber, H.G., & McKoy, K.F. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, **28**(4), 487–496.

Teufel, S., & Moens, M. 1997. *Sentence extraction as a classification task*. In Mani and Maybury (1997).

Teufel, S., & Moens, M. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, **28**(4), 409–445.